(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 050 802 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
08.11.2000 Bulletin 2000/45

(51) Int Cl.7: G06F 9/38, G06F 9/30

(21) Application number: 99410052.7

(22) Date of filing: 03.05.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: STMicroelectronics SA
94250 Gentilly Cedex (FR)

(72) Inventors:
- Wojcieszak, Laurent
38240 Meylan (FR)
- Cofler, Andrew
38340 Voreppe (FR)

(74) Representative: Driver, Virginia Rozanne
Page White & Farrer
54 Doughty Street
London WC1N 2LS (GB)

(54) Instruction prefetch unit

(57) A prefetch buffer is described which supports a computer system having a plurality of different instruction modes. The number of storage locations which are read out of the prefetch buffer during each machine cycle is controlled in dependence on the instruction mode.

Thus the prefetch buffer allows a number of different instruction modes to be supported and hides memory access latency.

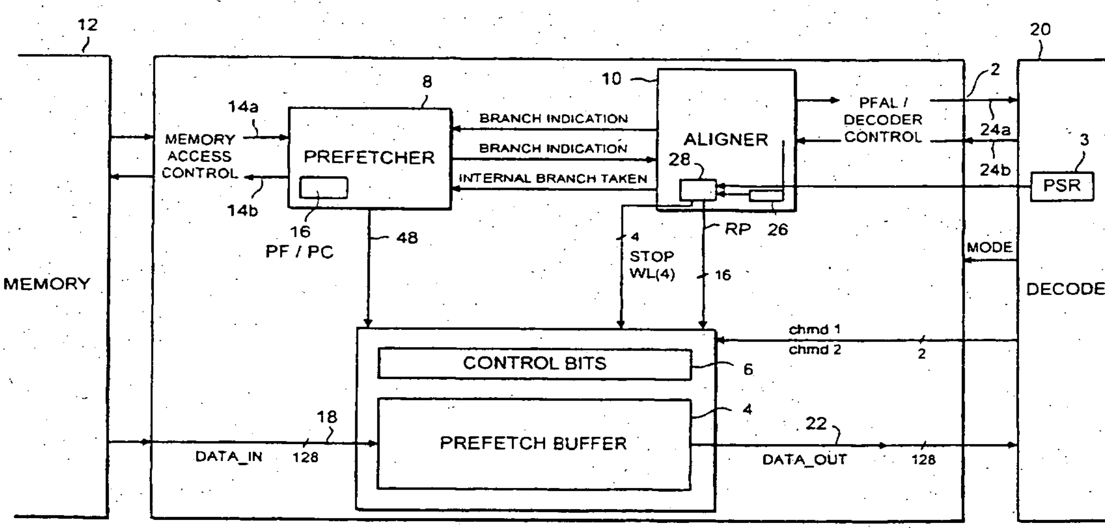

FIG. 1

EP 1 050 802 A1

Description

[0001] The present invention relates to a prefetch unit for use in a computer system.

[0002] In a computer system, instructions are typically fetched from a program memory, decoded and supplied to an execution unit where they are executed to run the program stored in the program memory. If more than one execution unit is provided, it is possible to arrange for very high speed instruction execution. In order to take advantage of this, it is clearly necessary to be able to supply decoded instructions to the execution unit at a sufficient rate. Presently, access times to memory cannot match execution speeds, and therefore several machine cycles are needed to access each new instruction from memory. Thus, there can be a severe performance degradation because the fetches from memory cannot match the rate at which instructions can be executed by the execution units.

[0003] According to the present invention there is provided a prefetch buffer for holding instructions in a processor having a memory and an instruction decode unit, the prefetch buffer comprising:

- a plurality of storage locations, each having the same bit capacity (2n bits) and arranged in groups with the same number p of storage locations in each group;
- a write port for selectively writing words of bit length n x p from the memory into respective groups of the prefetch buffer;
- read circuitry for reading instructions out of the prefetch buffer in dependence on an instruction mode of the processor, said instruction mode controlling the number of storage locations which are read during a machine cycle; and
- means for indicating when all storage locations in a group have been read so that a fetch signal can be issued to fetch a next word from the memory into the storage locations of that group.

[0004] In the described embodiment, each storage location has a capacity of 32 bits, and are arranged in groups of four such that each group has a capacity for a 128 bit word read out of memory on a memory fetch. In the described embodiment, four groups of storage locations are provided in the prefetch buffer, thus allowing for up to four successive memory accesses even if the first word has not yet been either received or executed. Moreover, because the processor supports more than one instruction mode, the time which it takes to read all storage locations in a group in terms of machine cycles can vary. According to the invention, the indicating means allow for a next word to be fetched from memory when all storage locations in a group have been read, however many machine cycles that has taken. Thus, memory latency is hidden through this mechanism.

[0005] According to a first instruction mode, one storage location is read out during each machine cycle to provide a pair of 16 bit instructions to the decode unit (referred to herein as GP16 mode).

[0006] According to a second instruction mode, two storage locations are read during each machine cycle to provide two 32 bit instructions to the decode unit (referred to herein as GP32 mode).

[0007] According to a third instruction mode, four storage locations are read out during each machine cycle to provide four instructions each of 32 bits to the decode unit (referred to herein as VLIW (Very Long Instruction Word) mode).

[0008] In the described embodiment, the indicating means comprises a set of flags, each group having a flag associated therewith which is set to indicate that all storage locations in the associated group have been read so as to initiate a subsequent memory fetch.

[0009] The invention also provides a prefetch unit comprising a prefetch buffer as hereinabove defined and control circuitry arranged to monitor the indicating means and to issue a fetch signal to memory to fetch the next word into the prefetch buffer when all storage locations in a group have been read. The control circuitry can include an aligner for controlling a read pointer determining the storage locations to be read in a next machine cycle.

[0010] For a better understanding of the present invention and to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings in which:-

- Figure 1 is a block diagram of a prefetch unit;
- Figure 2 illustrates the different instruction modes of the processor;
- Figure 3 illustrates the organisation of a prefetch buffer; and
- Figure 4 is a schematic diagram illustrating the operation of the prefetch buffer.

[0011] Figure 1 is a block diagram of a prefetch unit 2 for a processor, the prefetch unit 2 comprising a prefetch buffer 4 with associated control bits 6 and control circuitry comprising a prefetcher 8 and an aligner 10. The prefetcher 8 is connected to a program memory 12 and is responsible for initiating memory accesses to the program memory 12 using memory access control signals 14a,14b. The address in memory to which a fetch is initiated is held in a prefetch program counter 16 in the prefetcher 8. Control of the prefetch program counter is not discussed herein, but it can be assumed that fetches are initiated from memory in accordance with a sequence of instructions to be executed by the processor. That is, the prefetch program counter may be incremented each time as a sequence of adjacent instructions is fetched, or it may change according to branches, traps, interrupts etc. Responsive to a memory fetch initiated by the prefetcher, instruction words are supplied from the program memory 12 to the prefetch buffer 4 as rep-

resented by data-in path 18.

[0012] The aligner 10 controls reading of instructions from the prefetch buffer to a decoder 20 along data-out path 22. To do this, the aligner issues and is responsive to prefetcher align (PFAL)/decoder control signals 24a, 24b. The aligner 10 has an align program counter 26 which keeps track of how many instructions have been dispatched to the decoder 20 in each machine cycle, and a state machine 28 which generates a read pointer RP for controlling the prefetch buffer in a manner which is described in more detail hereinafter.

[0013] Instructions in the program memory 12 can have a length of 16 bits or 32 bits. The prefetch buffer supports three different instruction modes as described with reference to Figure 2 as follows. The instruction mode is held in a process status register (PSR) 3 and can be changed. Change mode signals chmd1,chmd2 are issued by the decoder 20 responsive to a change in instruction mode.

[0014] According to a first instruction mode, a pair of 16 bit instructions are supplied during each machine cycle to the decoder 20 from the prefetch buffer 4. This pair is denoted slot0 slot1 in bit sequences w0,w1 etc. This is referred to herein as GP16 mode.

[0015] According to a second instruction mode, two instructions each having a length of 32 bits are supplied to the decoder from the prefetch buffer in each machine cycle, for example w0 w1 in CYCLE 0. This mode is referred to herein as GP32 mode.

[0016] According to a third instruction mode, four instructions w0 w1 w2 w3 each of 32 bits in length are supplied to the decoder in each machine cycle. This is referred to herein as VLIW.

[0017] In all modes, each fetch operation initiated to the program memory 12 retrieves an instruction word of 128 bits in length.

[0018] Thus, in GP16 mode, the instruction word comprises eight 16 bit instructions, paired as slot0,slot1 for each machine cycle. In GP32 and VLIW mode, the instruction word comprises four 32 bit instructions.

[0019] The organisation of the prefetch buffer 4 is illustrated in Figure 3. In diagrammatic terms, the prefetch buffer can be considered to have four successive lines L0 to L3 each having a capacity of 128 bits. There is a single write port WPO having a width of 128 bits which receives data from the program memory via the data-in path 18 and an input latch FF-in and writes it into the selected line under the control of a write pointer WP [3 0]. Each line comprises four storage locations each having a capacity of 32 bits and each of which is shown diagrammatically divided into two 16 bit sections for the purposes of explanation. The storage locations are denoted F0 to F15. Each line in Figure 3 is referred to herein as a group of storage locations and has the capacity for one 128 bit line from memory. This allows up to four successive memory accesses to be made, even if the first instruction word has not been received or executed by the processor. While the instruction word in storage locations F0 to F3 is being decoded and subsequently executed, memory fetches can continue to be implemented into the storage locations F4 to F7, F8 to F11 and F12 to F15 until the buffer is full. By the time that a memory fetch has been made into the last group F12 to F15, it is most likely that the first group F0 to F3 will have been completely read out into the decoder and will thus be ready to receive a subsequent instruction word from memory. The number of cycles required to decode an instruction word in each group varies depending on the instruction mode of the machine in a manner which will be described in more detail in the following. Nevertheless, a minimum of one cycle is required for reading and decoding, and therefore the use of the prefetch buffer hides memory latency.

[0020] In order to save a cycle when the prefetch buffer is empty or flushed after a branch, data can bypass the prefetch buffer through a bypass circuitry BS. As described in more detail later, the bypass circuitry is implemented as a plurality of multiplexors (MUX0 to MUX3 in Figure 4).

[0021] Figure 4 is a more detailed diagram of the prefetch buffer and its associated read circuitry. The storage locations F0 to F15 are illustrated aligned vertically for the purposes of explanation.

[0022] The control bits 6 described above in Figure 1 include empty flags EF1 to EF4 which indicate when a complete 128 bit line of storage locations is empty such that a subsequent memory fetch can be initiated. When a fetch is instituted from memory, and data has been received by the prefetch buffer, the empty flag is cleared to indicate that those storage locations are now full.

[0023] Reading from the prefetch buffer will now be described with reference to the schematic diagram of Figure 4. The prefetch buffer includes four read ports RP1,RP2,RP3 and RP4. These read ports each take the form of multiplexors each capable of connecting selected ones of the storage locations F0 to F15 to a 32 bit output, pf-buf-out1,2,3 or 4. However, the read ports are not identical. The first read port RP1 has sixteen inputs each of which is connected to a respective storage location F0 to F15 and each of which can be connected to the output pf-buf-out1. The second read port RP2 has eight inputs which are respectively connected to storage locations F1,F3,F5,F7,F9,F11,F13,F15 to selectively connect the contents of those storage locations to the output pf-buf-out2.

[0024] The third read port RP3 has four inputs connected to storage locations F2,F6,F10 and F14 for selectively connecting the contents of those storage locations to the output pf-buf-out3. The fourth read port RP4 also has four inputs which are connected to storage locations F3,F7,F11 and F15 for selectively connecting the contents of those storage locations to the output pf-buf-out4.

[0025] The read ports RP1 to RP4 are controlled by the read pointer RP from the aligner 10 in dependence on the instruction mode of the machine and the conse-

quential number of machine cycles required for decoding each instruction word.

[0026] Alternatively, for instructions supplied directly from memory along data-in path 18, the control of instructions supplied to the decoder in dependence on the instruction mode and machine cycles is additionally controllable by multiplexors MUX0,MUX1,MUX2 and MUX3. These receive at their input respective bits of the 128 bit data-in path 18 to supply a 32 bit sequence to each multiplexor in each machine cycle as described in the following.

[0027] The selection of which instructions within the instruction word are supplied to the decoder 20 is made on dependence on the instruction mode as described in the following. In Figure 3, the symbols w0 to w3 are used on different input lines of the multiplexors MUX0 to MUX3 to represent different 32 bit sequences, as in Figure 4. The definition of each 32 bit sequence depends on the instruction mode, but bits of the data-in path are always allocated as w0 [0:31], w1 [32:63], w2 [64:95], w3 [96:127]. The inputs to the multiplexors are individually labelled so as to distinguish between them. That is, in GP16 mode, on the first decode cycle, cycle 0, the first sequence w0 is supplied to the decoder 20. This presents a pair of 16 bit instructions, slot0,slot1 (w0) for simultaneous decoding by the decoder 20. On the next cycle, cycle 1, the sequence w1 is supplied, presenting the next pair of 16 bit instructions slot0,slot1 (w1) for decoding. In GP16 mode, the read port RP1 and the multiplexor MUX0 are the only read devices which are used and the control of the word which is supplied to the decoder is made by the multiplexor MUX0 under the control of signal mux-ctrl0, and the read pointer RP. If the signal mux-ctrl0 selects the read port output pf-buf-out1, the read pointer selects inputs F0 to F3 over four successive cycles CYCLE0 to CYCLE3 to read out successively w0 to w4. Once storage location F3 has been read out, the read port counter will reset the read port RP1 so that it reads out from storage locations F4 to F7 over the next four cycles. If the buffer is not in use, the first instruction pair w0 is read out by the multiplexor MUX0. That is in cycle 0, input M00 of the multiplexor MUX0 is selected. Meanwhile, the 128 bit line is loaded into the first location of the prefetch buffer and the read pointer points to the next location to be read out by the decoder. Therefore on cycle 1, the next instruction pair w1 is read out by the multiplexor MUX0 by selecting pf-buf-out1

[0028] In GP32 mode, in the first machine cycle the first two instructions w0,w1 are presented to the decoder 20. In the subsequent cycle, cycle 1, the next two instructions w2,w3 are presented to the decoder. This utilises read ports RP1 and RP2 and the multiplexors MUX0 and MUX1. If the signal mux-ctr10 is set to pf-buf-out1, and mux-ctrl1 to pf-buf-out2, then the read pointer RP is set to F0 for RP1 and F1 for RP2 in cycle 0. In cycle 1, it is changed to F2 and F3 respectively. Instructions are then read over the next two cycles from the next group of storage locations F4 to F7 by altering the setting of the read ports RP1 and RP2 responsive to the read pointer RP. Alternatively, when read from the data-in path 18, in the first cycle, the first input M10 of the multiplexor MUX1 is set to read w1 (bits 31 to 63) and the first input M00 of the multiplexor MUX0 is set to read w0 (bits 0 to 31). Thus, instructions w0 and w1 are presented to the decoder 20 in CYCLE 0. Meanwhile, the 128 bit line is loaded into the prefetch buffer so that in the subsequent cycle, CYCLE 1, w2 and w3 are read from the buffer by selecting pf-buf-out1 and pf-buf-out2.

[0029] In VLIW mode, four 32 bit instructions W0 to W3 (slot0 to slot3) are supplied simultaneously to the decoder 20 in each machine cycle, e.g. CYCLE 0. The multiplexors MUX2 and MUX3 are set according to the control signals mux-ctrl2 and mux-ctrl3 respectively to allow the instruction words w2 and w3 to be read either from the buffer or from the data-in path 18. In other respects, the settings of RP1 and RP2, MUX0 and MUX1 are as in GP32 mode. However, in the subsequent cycle, e.g. CYCLE 1 in VLIW mode, it will be noticed that the instruction words w2 and w3 which would have been remaining in GP32 mode have now been read out. Therefore, the read pointer RP can immediately move on to the next set of storage locations F4 to F7 to read out the subsequent VLIW instruction word containing the next four instructions.

[0030] Data is passed from the multiplexors MUX0 to MUX3 to respective output flip-flops FF0 to FF3 via a set of control gates labelled GC1, GC2 and GS0 to GS3. The control gates GC1,GC2 are responsive to change mode signals chmd1,chmd2 respectively which indicate to the prefetch unit that there has been a change in the instruction mode in which the machine is operating. The control gates GS0 to GS3 are responsive to respective stop signals stop[0] to stop[3] to prevent any new data from entering the decoder from that output flip-flop. These effectively allow the decoder to be stalled. In a stop condition, the outputs of the flip-flops are recirculated to the input of its associated control switch to prevent unnecessary operation of the subsequent decoder.

[0031] Operation of the prefetch unit responsive to the change mode signals chmd1 and chmd2 will now be described. The output flip-flop FF0 is connected to a single 32 bit decoder and to two 16 bit decoders. When the machine is in GP16 mode, the outputs of the two 16 bit decoders are selected for the instruction pair supplied to the flip-flop FF0. When the machine is in GP32 mode, the output of the 32 bit decoder is selected. The remaining flip-flops FF1 to FF3 are each connected to respective 32 bit decoders.

[0032] A first change mode signal chmd1 signals a change of machine instruction mode from GP32 to GP16. If the machine had been operating in GP32 mode, consider the situation at the end of cycle 0 which reference to Figure 2. Instructions w0 and w1 will have been supplied via the flip-flops FF0 and FF1 to the respective 32 bit decoders of the decoder 20. However,

the change in instruction mode now implies that the 32 bit sequence which was formerly to be considered as the second instruction W1 in cycle 0 of GP32 mode, in fact contains a pair of 16 bit instructions as denoted in cycle 1 of GP16 mode. Thus, the output of the 32 bit decoder connected to the flip-flop FF1 needs to be ignored, and the 32 bit sequence w1 needs to be reapplied to the two 16 bit decoders connected to the output flip-flop F0. This is achieved by the recirculation line 42 from the output of the flip-flop FF1 to the input of the control gate CG1.

[0033] Conversely, control signal chmd2 denotes a change of instruction mode from GP16 to GP32. Consider again the effect at the end of cycle 0 with reference to Figure 2. The instruction pair denoted w0 has just been decoded in GP16 mode, and the expectation is that the machine will now wait for the next instruction pair w1. However, in GP32 mode, that word w1 represents a single instruction and the change mode signal chmd2 allows it to be applied directly through the control gate GC2 to the output flip-flop FF1 so that it can be applied directly to the input of the dedicated 32 bit decoder connected to the output of the flip-flop FF1. This allows the instruction w1 to be decoded as a single 32 bit instruction. In the next cycle, instructions w2 and w3 can be transmitted normally as indicated by cycle 1 in GP32 mode in Figure 2.

[0034] It will be clear from the above that the number of cycles needed to read all four storage locations in a group depends on the instruction mode. That is, in GP16 mode, four cycles are needed, in GP32 two cycles are needed and VLIW one cycle is needed. When all the storage locations F0 to F3 in the first group have been read, the first empty flag EF1 is cleared to empty.

[0035] The aligner controls the setting and clearing of the "empty" flags using information from the read pointer. The aligner detects when the read pointer goes from one line (128 bits) to the next. When this occurs, the "empty" flag corresponding to the page which has just been read is set.

[0036] The state of an empty flag being cleared is detected by the prefetcher 8 along line 48 and a fetch is initiated to the next prefetch address in the prefetch program counter 16. Thus, the next instruction line is fetched from memory and the write pointer WP is set to write it into storage locations F0 to F3. In the meantime, the read pointer has moved to the second group F4 to F7 to read and decode instructions of that group. When those storage locations are empty, the empty flag EF2 is cleared, a next memory fetch is initiated by the prefetcher 8 and the read pointer moves onto the group F8 to F11. As can readily be seen, the prefetch buffers masks a latency of memory fetches of at least three cycles in the VLIW mode, and a greater number of cycles in GP32 and GP16 mode. Signals are supplied from the decoder along line 24b to the aligner 10 indicating what mode the decoder is operating in so that the aligner can adjust the align program counter 26 accordingly and keep track of the next instructions to be decoded so that the read pointer RP can correctly be issued by the state machine 28.

Claims

1. A prefetch buffer for holding instructions in a processor having a memory and an instruction decode unit, the prefetch buffer comprising:

   a plurality of storage locations, each having the same bit capacity (2n bits) and arranged in groups with the same number p of storage locations in each group;
   a write port for selectively writing words of bit length n x p from the memory into respective groups of the prefetch buffer;
   read circuitry for reading instructions out of the prefetch buffer in dependence on an instruction mode of the processor, said instruction mode controlling the number of storage locations which are read during a machine cycle; and
   means for indicating when all storage locations in a group have been read so that a fetch signal can be issued to fetch a next word from the memory into the storage locations of that group.

2. A prefetch buffer according to claim 1, wherein the read circuitry is responsive to a first instruction mode to read out one storage location during each machine cycle to provide two instructions each of bit length n to the decode unit (GP16).

3. A prefetch buffer according to claim 1 or 2, wherein the read circuitry is responsive to a second mode of operation to read out two storage locations during each machine cycle to provide two instructions each of bit length 2n to the decode unit (GP32).

4. A prefetch buffer according to claim 1 or 2, wherein the read circuitry is responsive to a third mode of operation to read out four storage locations during each machine cycle to provide four instructions each of bit length 2n to the decode unit (VLIW).

5. A prefetch buffer according to any preceding claim, wherein said indicating means comprises a set of flags, each group having a flag associated therewith which is set to indicate that all storage locations in the associated group have been read.

6. A prefetch buffer according to any preceding claim, wherein the number of storage locations in the prefetch buffer matches the latency of a memory fetch operation.

7. A prefetch unit comprising:

a prefetch buffer according to any preceding claim; and
control circuitry arranged to monitor said indicating means and to issue a fetch signal to memory to fetch said next word into the prefetch buffer when all storage locations in a group have been read.

8. A prefetch unit according to claim 7, wherein the control circuitry includes an aligner for controlling a read pointer determining the storage location to be read in the present machine cycle.

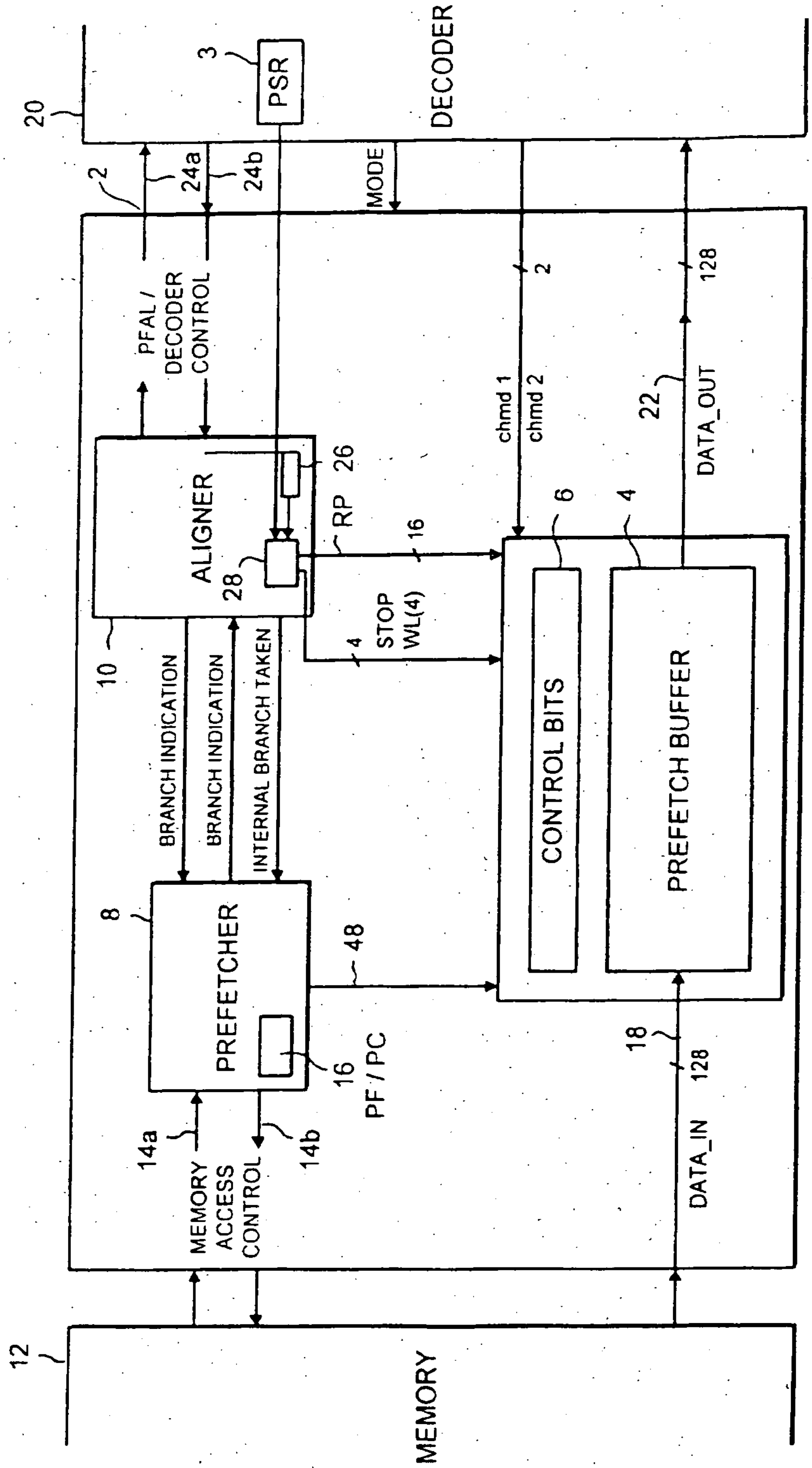
DECODER
20
3
PSR
2
24a
24b
MODE
PFAL / DECODER CONTROL
chmd 1
chmd 2
2
22
DATA_OUT
128
ALIGNER
26
RP
28
16
STOP WL(4)
4
6
4
10
BRANCH INDICATION
BRANCH INDICATION
INTERNAL BRANCH TAKEN
CONTROL BITS
PREFETCH BUFFER
8
PREFETCHER
48
16
PF / PC
14a
14b
MEMORY ACCESS CONTROL
18
128
DATA_IN
12
MEMORY

FIG. 1

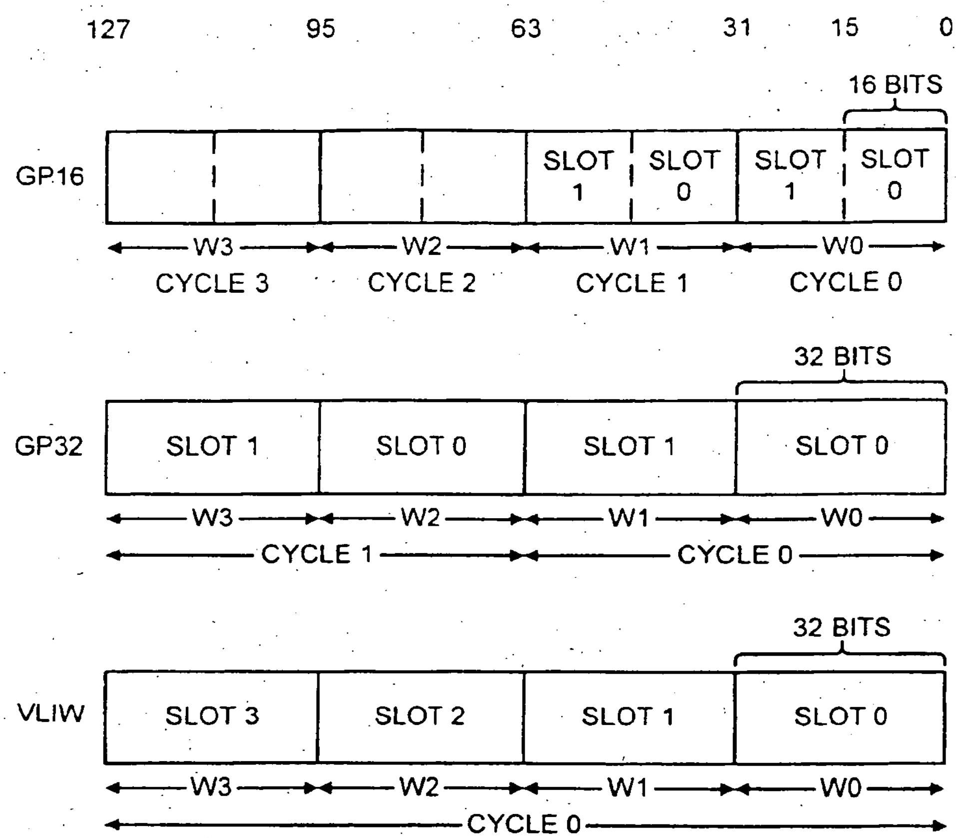
127
95
63
31
15
0
16 BITS
GP16
SLOT 1
SLOT 0
SLOT 1
SLOT 0
W3
W2
W1
W0
CYCLE 3
CYCLE 2
CYCLE 1
CYCLE 0
32 BITS
GP32
SLOT 1
SLOT 0
SLOT 1
SLOT 0
W3
W2
W1
W0
CYCLE 1
CYCLE 0
32 BITS
VLIW
SLOT 3
SLOT 2
SLOT 1
SLOT 0
W3
W2
W1
W0
CYCLE 0

FIG. 2

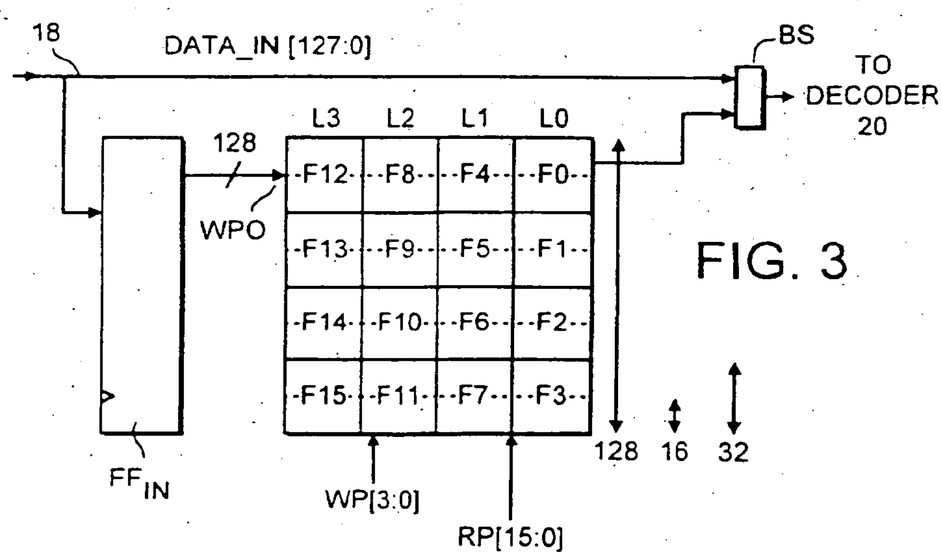
18
DATA_IN [127:0]
BS
TO DECODER 20
L3
L2
L1
L0
128
WPO
F12
F8
F4
F0
F13
F9
F5
F1
F14
F10
F6
F2
F15
F11
F7
F3
FF IN
WP[3:0]
RP[15:0]
128
16
32

FIG. 3

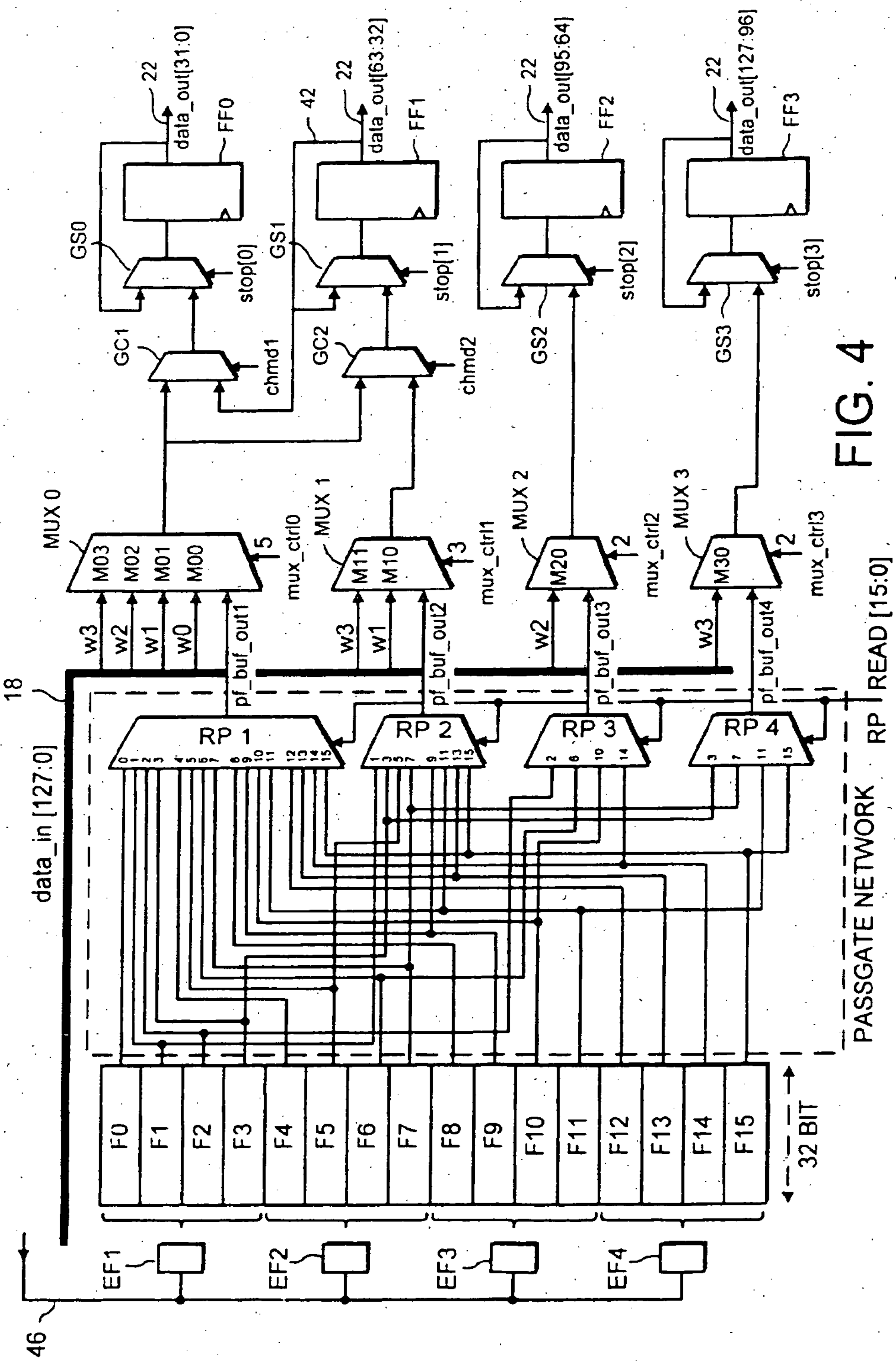
data_out[31:0]
data_out[63:32]
data_out[95:64]
data_out[127:96]
22
42
FF0
FF1
FF2
FF3
GS0
GS1
GS2
GS3
stop[0]
stop[1]
stop[2]
stop[3]
GC1
GC2
chmd1
chmd2
MUX 0
MUX 1
MUX 2
MUX 3
M03
M02
M01
M00
M11
M10
M20
M30
mux_ctrl0
mux_ctrl1
mux_ctrl2
mux_ctrl3
5
3
2
w3
w2
w1
w0
pf_buf_out1
pf_buf_out2
pf_buf_out3
pf_buf_out4
18
data_in [127:0]
RP 1
RP 2
RP 3
RP 4
RP
READ [15:0]
PASSGATE NETWORK
F0
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
F13
F14
F15
32 BIT
EF1
EF2
EF3
EF4
46

FIG. 4

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 41 0052

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	EP 0 436 341 A (MOTOROLA INC) 10 July 1991 (1991-07-10) * page 3, line 43 - page 4, line 44 * * page 7, line 52 - page 8, line 2 * ---	1,7	G06F9/38 G06F9/30
A	EP 0 417 013 A (FUJITSU LTD) 13 March 1991 (1991-03-13) * column 2, line 50 - column 3, line 23 * * column 4, line 30 - line 38 * ---	1,7	
A	WO 99 14669 A (SONY ELECTRONICS INC) 25 March 1999 (1999-03-25) * page 10, line 16 - line 21 * ---	2-4	
A	US 4 654 781 A (SCHWARTZ MARTIN J ET AL) 31 March 1987 (1987-03-31) * the whole document * ---	5,7,8	
A	US 5 717 881 A (BLEWETT RICHARD G ET AL) 10 February 1998 (1998-02-10) * column 6, line 19 - line 36 * * column 27, line 53 - line 61 * * figure 5A * * figure 5B * -----	1	TECHNICAL FIELDS SEARCHED (Int.Cl.7) G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	24 November 1999	Moraiti, M

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03 82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 41 0052

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

24-11-1999

Patent document cited in search report		Publication date	Patent family member(s)			Publication date
EP 0436341	A	10-07-1991	DE	69030648	D	12-06-1997
			DE	69030648	T	13-11-1997
			JP	2949857	B	20-09-1999
			JP	5324317	A	07-12-1993
			US	5317701	A	31-05-1994
EP 0417013	A	13-03-1991	JP	3095629	A	22-04-1991
			US	5581774	A	03-12-1996
WO 9914669	A	25-03-1999	AU	9388398	A	05-04-1999
US 4654781	A	31-03-1987	CA	1183275	A	26-02-1985
			DE	3236524	A	28-04-1983
			GB	2108737	A,B	18-05-1983
			JP	58086651	A	24-05-1983
US 5717881	A	10-02-1998	US	5640524	A	17-06-1997
			US	5430884	A	04-07-1995
			US	5197130	A	23-03-1993
			US	5598547	A	28-01-1997
			US	5706490	A	06-01-1998
			US	5623650	A	22-04-1997
			US	5745721	A	28-04-1998
			US	5659706	A	19-08-1997
			US	5544337	A	06-08-1997
			AU	7151991	A	24-07-1991
			CA	2071481	A	30-06-1991
			DE	69033272	D	07-10-1999
			EP	0512007	A	11-11-1992
			WO	9110194	A	11-07-1991
			US	5499356	A	12-03-1996
			US	5381536	A	10-01-1995
			US	5561784	A	01-10-1996
			US	5524255	A	04-06-1996
			US	5208914	A	04-05-1993
			US	5168547	A	01-12-1992
			US	5239629	A	24-08-1993
			US	5175862	A	29-12-1992
			US	5165038	A	17-11-1995
			US	5168570	A	01-12-1992
			US	5179702	A	12-01-1993
			US	5193187	A	09-03-1993

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82